UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/091,237	03/04/2002	Hong Su	10013661-1	7188

7590 04/26/2006

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
STEVENS, ROBERT	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 04/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/091,237	SU ET AL.
	Examiner	Art Unit
	Robert Stevens	2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: amendment filed 12/15/2005.

2. This action is **FINAL**.

3. The Office maintains the previous rejections of the claims under 35 U.S.C. §103(a), in light of the arguments presented in the amendment filed 12/15/2005.

4. Claims 1-26 are pending. Claim 1 is independent.

Office Notes

5. The Request for Continued Examination (RCE) filed 12/15/2005 is considered improper because prosecution in the application was not closed at the time of the filing of the RCE. It is noted that the Office action mailed 11/28/2005 placed the application in a non-final status. This communication responds to the amendment/remarks submitted with the RCE in accordance with the procedures set forth by MPEP §706.07(h) III. A. 1., concerning the treatment of an improper RCE when prosecution is not closed.

6. The declarations under 37 CFR 1.131 are considered moot in light of the fact that the art to which these declarations were directed was not the subject of the last Office Action.

Specification

7. The specification is objected to because a Federally Sponsored Research and Development statement, as set forth in MPEP §310, appears to be necessary. On page 29 of the Power Point briefing [Su, Hong, et al., "Automating Transformation of XML Documents", WIDM 2001 Powerpoint Presentation, Atlanta, Ga, Nov. 2001, pp. 1-29] presented at the WIDM 2001 Conference by the inventors, mention is given to the involvement of the NSF (i.e., a US Government agency that provides R&D funding to researchers). Note that this briefing (a copy of which was provided with the Office action mailed 11/28/2005 and cited in the 11/28/2005 Form PTO-892) discloses the subject matter of the WIDM 2001 paper that is the subject of the inventors' Rule 131 affidavits concerning reduction to practice.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. **Claims 1, 3-4 and 10-16 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM").

Independent claim 1 states:

A method of document transformation comprising:
a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;
b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and
c) generating a sequence of transformation operations that transforms said source tree to said target tree.

Regarding these limitations ...

a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;
b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the element graphs for the customer and client DTDs, and section "2.3 Construction of a Directed Acyclic Graph (DAG)" describes the process of creating the DAG, or tree, data structure.

c) generating a sequence of transformation operations that transforms said source tree to said target tree.

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

Regarding dependent claim 3, SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section "3.1 Initial Leaf Vertices Matching" on pages 5-6, discussing operational steps for "Matching Criterion 1" between leaf nodes. Discussion of "Matching Criterion 2" in section "4 Detection of Hierarchically Equivalent Elements" on page 7, discloses

a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched.

Regarding dependent claim 4, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

Independent claim 10 states:

A method of document transformation comprising:
a) modeling a source schema of XML and a target schema of XML as a tree structure creating source tree and a target tree, said source tree having a plurality of source nodes, said target tree having a plurality of target nodes; and
b) generating a sequence transformation operations that transforms said source XML document to said target XML document, wherein said plurality of source nodes of said source schema are matched and transformed to said plurality of target nodes in said target schema.

Regarding these limitations ...

a) modeling a source schema of XML and a target schema of XML as a tree structure creating source tree and a target tree, said source tree having a plurality of source nodes, said target tree having a plurality of target nodes; and
wherein said plurality of source nodes of said source schema are matched and transformed to said plurality of target nodes in said target schema.

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the element graphs for the customer and client DTDs, and section "2.3 Construction of a Directed Acyclic Graph (DAG)" describes the process of creating the DAG, or tree, data structure. SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section "3.1 Initial Leaf Vertices Matching" on pages 5-6, discussing operational steps for "Matching Criterion 1" between leaf nodes. Discussion of "Matching Criterion 2" in section "4 Detection of Hierarchically Equivalent Elements" on page 7, discloses a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched. SchemaMatching also discloses three exemplary methods of transforming DTD elements on page 3.

b) generating a sequence transformation operations that transforms said source XML document to said target XML document,

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

Regarding dependent claim 11, SchemaMatching discloses matching nodes and selecting a best matching plan in phases numbered 3 and 4 on page 2 (immediately above the section labeled "2 DTD Data Model"). SchemaMatching also discloses distance (i.e., cost) and

element overlap (i.e., data loss) in Example 4 of page 7, and the paragraph preceding Example 4, which discuss computing a number reflective of the amount of overlap of two graphs.

Regarding dependent claim 12, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as

taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

Regarding dependent claim 13, SchemaMatching discloses the use of source and target DTDs in phase number "1." above the section entitled "2 DTD Data Model", discussing modeling source and target DTDs as graphs.

Regarding dependent claim 14, SchemaMatching discloses simplifying DTDs into a reduced DTD in section "2.1 Simplification Transformation on DTD", discussing, for example, a transformation that folds a group into a flattened representation.

Regarding dependent claim 15, SchemaMatching discloses simplifying DTDs into a reduced DTD in section "2.1 Simplification Transformation on DTD", discussing, for example, a transformation that merges sub-elements having the same name in a content model.

Regarding dependent claim 16, SchemaMatching does not explicitly disclose constraining node operations. XEM, though, teaches the well-known use of quantifier node notation, such as qmark "?", asterisk "*" and plus "+" in sub-section 2. Constraint Node" on page 3. XEM further discusses labelling in the third paragraph bleow the section header "2.2 The DTD Data Model", discussing the labelling function *l* (i.e., "ell"). XEM discloses on pages 3-6 various operations performed on DTD models. In section "2.2 The DTD Data Model", XEM

sets forth how nodes are labelled and the notation used by one skilled in the art at the time of the invention. Section "3.2 DTD Change Primitives" sets forth various operations using that notation, including folding or flattening (see page 5 "5. flattenGroup(E, pos)") [it being obvious that one performed the inverse of folding in order to unfold], and the changing of attributes [including relabeling] in section "3.2.3 Changes to an Attribute Type Definition" on page 6.

When certain functions are performed is merely an obvious variant.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

10. **Claims 2 and 17-21 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Swamy et al. (US Patent No. 6,874,141, filed Jun. 29, 2000 and issued Mar. 29, 2005, hereafter referred to as "Swamy").

Regarding dependent claim 2, SchemaMatching does not explicitly disclose the generation of XSLT stylesheets. Swamy, though, teaches generation of XSLT stylesheets in the Abstract, discussing the generation of XSL code representation (and an XSLT stylesheet) of a mapping between a source and a target schema.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Swamy for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to compile a graphical representation of data transformations into an XSL stylesheet representation of the mapping, as taught by Swamy in col. 3 lines 19-25. These references were all applicable to the same field of endeavor, i.e., XML programming.

Claim 17 is substantially similar to claim 2, and therefore likewise rejected.

Independent claim 18 states:

A computer system comprising:
a processor; and
a computer readable memory coupled to said processor and containing program instructions that, when executed, implement a method of document transformation comprising:
a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;
b) modeling target XML document corresponding to a target schema as a target tree having a plurality of \target nodes; and
c) generating a sequence of transformation operations that transforms said source tree said target tree.

Regarding these limitations ...

a) modeling source XML document corresponding to a source.schema as a source tree having a plurality of source nodes;
b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the element graphs for the customer and client DTDs, and section "2.3 Construction of a Directed Acyclic Graph (DAG)" describes the process of creating the DAG, or tree, data structure.

c) generating a sequence of transformation operations that transforms said source tree to said target tree.

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have

allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

a processor; and
a computer readable memory coupled to said processor and containing program instructions that, when executed, implement a method of document transformation comprising;

SchemaMatching does not explicitly disclose a hardware implementation environment. Swamy, though, teaches a hardware implementation environment in Figure 15, disclosing a processor (#521) and memory units (#522, 527, 529, 531) coupled via a bus (#523) for executing applications (#536), including source to target schema transformations (Abstract).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Swamy for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to compile a graphical representation of data transformations into an XSL stylesheet representation of the mapping, as taught by Swamy in col. 3 lines 19-25. These references were all applicable to the same field of endeavor, i.e., XML programming.

Claim 19 is substantially similar to claim 2, and therefore likewise rejected.

Regarding dependent claim 20, SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section “3.1 Initial Leaf Vertices Matching” on pages 5-6, discussing operational steps for “Matching Criterion 1” between leaf nodes. Discussion of “Matching Criterion 2” in section “4 Detection of Hierarchically Equivalent Elements” on page 7, discloses

a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched.

Regarding dependent claim 21, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section "4 Completeness of DTD Change Operations" on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Swamy, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

11. **Claims 5-9 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Peter Buneman et al ("UnQL: A Query Language and Algebra for SemiStructured Data Based on Structural Recursion", The VLDB Journal, Issue No. 9, Springer-Verlag, © 2000, pp. 76-110, hereafter referred to as "Buneman").

Regarding dependent claims 5-6 and 8-9, SchemaMatching discloses matching source and target nodes and generating a transformation between nodes in phases 3 and 4 on page 2, discussing the matching likelihood of component pairs" (phase 3) and producing a "best matching plan" in phase 4. However, SchemaMatching does not explicitly disclose computing for a sequence of transformation operations, such as unfolding. Buneman, though, teaches the calculation of value equivalence in performing operations on trees in section "4.1 value Equivalence" starting on page 88, discussing the determination of a value equivalence for a data graph d and the resulting graph from an unfold operation in the first and second paragraphs under the section heading.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Buneman for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to implement a value-based semistructured data model, as

taught by Buneman in last paragraph on page 77. These references were all applicable to the same field of endeavor, i.e., XML programming.

Regarding dependent claim 7, SchemaMatching does not explicitly disclose matching node labels. XEM, though, teaches the use of node labels in section "2.2 The DTD Data Model", discussing a labeling for node attributes. It was implicit in node matching that at least one node attribute must be matched, and it was a obvious variant at the time of the invention as to which attribute (e.g. label name) was matched between nodes.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Buneman, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

12. **Claims 22-26 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Swamy et al. (US Patent No.

6,874,141, filed Jun. 29, 2000 and issued Mar. 29, 2005, hereafter referred to as "Swamy") and Peter Buneman et al ("UnQL: A Query Language and Algebra for SemiStructured Data Based on Structural Recursion", The VLDB Journal, Issue No. 9, Springer-Verlag, © 2000, pp. 76-110, hereafter referred to as "Buneman").

Regarding dependent claims 22-23 and 25-26, SchemaMatching discloses matching source and target nodes and generating a transformation between nodes in phases 3 and 4 on page 2, discussing the matching likelihood of component pairs" (phase 3) and producing a "best matching plan" in phase 4. However, SchemaMatching does not explicitly disclose computing for a sequence of transformation operations, such as unfolding. Buneman, though, teaches the calculation of value equivalence in performing operations on trees in section "4.1 value Equivalence" starting on page 88, discussing the determination of a value equivalence for a data graph d and the resulting graph from an unfold operation in the first and second paragraphs under the section heading.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Buneman for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to implement a value-based semistructured data model, as taught by Buneman in last paragraph on page 77. These references were all applicable to the same field of endeavor, i.e., XML programming.

Regarding dependent claim 24, SchemaMatching does not explicitly disclose matching node labels. XEM, though, teaches the use of node labels in section "2.2 The DTD Data Model",

discussing a labeling for node attributes. It was implicit in node matching that at least one node attribute must be matched, and it was a obvious variant at the time of the invention as to which attribute (e.g. label name) was matched between nodes.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Buneman, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

Response to Arguments

13. Applicant's arguments have been fully considered but they are not persuasive.

Applicant argues on pages 10-15 that the rejection of the claims under 35 USC 102(a), as being anticipated by Jeong, and under 35 USC 103(a), as being unpatentable over Jeong in view of Geiger or Oracle9i, as appropriate, are improper.

The Office respectfully asserts that these arguments are moot in light of the withdrawal of these rejections in the previous action, mailed 11/28/2005.

For these reasons, the Office maintains/asserts the rejections under 35 USC 103(a) as set forth above.

14. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert Stevens whose telephone number is (571) 272-4102. The examiner can normally be reached on M-F 6:00 - 2:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather R. Herndon can be reached on (571) 272-4136. The current fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. Additionally, the main number for Technology Center 2100 is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert Stevens
Art Unit 2176
Date: April 17, 2006

rs

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER
4/24/2006